AMENDMENTS TO THE CLAIMS

Please amend the claims as follows.

1. (Currently Amended) A method for tracing an instrumented program on a processor having an x86 architecture, comprising:

triggering a probe in the instrumented program;

obtaining an original instruction associated with the probe, wherein obtaining the original instruction comprises searching a look-up table using a program counter value, wherein the look-up table comprises the original instruction associated with the probe and an address associated with the original instruction;

allocating a second scratch space for a second thread;

loading the original instruction into [[a]] the second scratch space, wherein the scratch space is allocated on a per-thread basis, and wherein a first scratch space for a first thread executing in the instrumented program was previously allocated;

loading a jump instruction for the x86 architecture into the second scratch space wherein the jump instruction includes a next program counter value;

executing the original instruction in the second scratch space using a second thread; and

executing the jump instruction in the scratch space using the second thread.

2. (Previously Presented) The method of claim 1, further comprising:

emulating the original instruction to determine the program counter value if the original instruction is a control-flow instruction; and

returning control to the thread at an address of the program counter value if the original instruction is a control-flow instruction.

3. (Previously Presented) The method of claim 1, further comprising:

determining the next program counter value by incrementing the program counter value using a size of the original instruction.

4. (Original) The method of claim 1, wherein the probe corresponds to a trap.

5. (Canceled)

6. (Canceled)

7. (Original) The method of claim 1, wherein the instrumented program is executed on a multi-thread architecture.

8. (Currently Amended) A system for tracing an instrumented program on a processor having an x86 architecture, comprising:

a first thread configured to execute the instrumented program;

a second thread configured to execute the instrumented program;

a look-up table arranged to store an address and a corresponding original instruction;

a trap handler configured to halt execution of the second thread when a trap instruction is encountered, use an address of the trap instruction to obtain the corresponding original instruction from the look-up table, and generate a jump instruction to an address in the instrumented program;

a first scratch space allocated for the first thread;

a second scratch space arranged to store the original instruction and the jump instruction, wherein the scratch space is allocated on a per-thread basis; and

an execution facility for executing the original instruction in the second scratch space to collect data and executing the jump instruction, wherein the execution facility is a processor based on the x86 architecture.

9. (Original) The system of claim 8, further comprising:

a buffer for storing the data.

10. (Original) The system of claim 8, further comprising:

a tracing framework configured to emulate the original instruction to determine a value of a program counter if the original instruction is a control-flow instruction and to return

control to a thread at an address of the program counter value if the original instruction is a control-flow instruction.

11. (Original) The system to claim 8, wherein the trap handler sets a destination of the jump instruction to a next address immediately following an address of the trap instruction.

12. (Canceled)

13. (Original) The system of claim 8, wherein the instrumented program is executed on multi-thread architecture.